



Optimal drone placement and cost-efficient target coverage

Dimitrios Zorbas, Luigi Di Puglia Pugliese, Tahiry Razafindralambo,
Francesca Guerriero

► To cite this version:

Dimitrios Zorbas, Luigi Di Puglia Pugliese, Tahiry Razafindralambo, Francesca Guerriero. Optimal drone placement and cost-efficient target coverage. Journal of Network and Computer Applications (JNCA), 2016, 75, pp.16 - 31. 10.1016/j.jnca.2016.08.009 . hal-01783842

HAL Id: hal-01783842

<https://hal.inria.fr/hal-01783842>

Submitted on 3 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal drone placement and cost-efficient target coverage

Dimitrios Zorbas ^{a,*}, Luigi Di Puglia Pugliese ^b, Tahiry Razafindralambo ^a,
Francesca Guerriero ^b

^a Inria Lille, Nord Europe, France

^b Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy

A B S T R A C T

Observing mobile or static targets in the ground using flying drones is a common task for civilian and military applications. We introduce the minimum cost drone location problem and its solutions for this task in a two-dimensional terrain. The number of drones and the total energy consumption are the two cost metrics considered. We assume that each drone has a minimum and a maximum observation altitude. Moreover, the drone's energy consumption is related to this altitude. Indeed, the higher the altitude, the larger the observed area but the higher the energy consumption. The aim is to find drone locations that minimize the cost while ensuring the surveillance of all the targets. The problem is mathematically solved by defining an integer linear and a mixed integer non-linear optimization models. We also provide some centralized and localized heuristics to approximate the solution for static and mobile targets. A computational study and extensive simulations are carried out to assess the behavior of the proposed solutions.

1. Introduction

During the recent years, we have witnessed an increased interest in using flying devices for monitoring applications. Emerging pervasive application systems, such as observation and tracking of unknown moving or static objects, will face a number of challenges, including the need to operate in extreme and unknown environments. It is important to develop reliable systems by providing the most appropriate and up to date information, at the lowest cost. The technology to observe this kind of systems, such as drones, has become increasingly prevalent and has many practical applications, including emergency or rescue operations, military operations, and environmental monitoring.

We assume drones equipped with one or more electrical motors (quadcopters) and a fixed-angle camera targeting on the ground. The drones are able to identify static or mobile ground targets, which are considered as points that have to be monitored, such as machines, animals, humans etc. We assume a binary sensing model, with isotropic sensing, and that the targets are moving on a flat and smooth area of interest. In case of mobile

targets, no a priori information about their mobility is known, except their maximum speed. Our objective is the *optimal* deployment of drones ensuring, at the same time, that each target is covered by at least one drone. Another dimension in our problem is that each drone can change its coverage radius, depending on its altitude that allows it to cover more or less targets. It is assumed that the energy consumed by each drone is related to its altitude. We take into consideration an empirical energy consumption model based on our own measurements with electrical motors and drone manufacturers data. The focus of this work is the minimization of the cost, that is the number of drones or the total energy consumption. The number of drones depends on the number of targets, their dispersion, and their movement. In this paper, we minimize the total cost only in order to simplify the computational model. However, a split rule can be easily added in heuristics to decrease altitudes – and thus the energy consumption – when there is such a need. Moreover, in our localized solution, drones are always keen to retain low altitudes even if the objective is the minimization of the number of nodes.

The optimal placement of a set of monitoring devices is a very challenging problem, which, in most of the cases, has been proven to be NP-Hard (Younis and Akkaya, 2008). It is then critical to design fast, efficient and autonomous algorithms to support pervasive, “any time, any place” services in these highly mobile environments, prone to time and space evolution. Furthermore, since the system is autonomous, energy becomes a major concern.

Hence, the energy reservation represents a fairly complex point of interest in coverage problems and constitutes a main contribution of this paper. To the best of our knowledge, this is one of the first papers that specifically deals with the cost minimization of static or mobile target tracking in the context of drones.

A simplified version of the considered problem has been addressed by the same authors in [Zorbas et al. \(2013\)](#), where it is assumed that an infinite number of drones is available to cover a set of mobile ground targets. In [Zorbas et al. \(2013\)](#), the objective is to minimize the total energy consumption. In the current paper, we handle a more general aspect of the problem under consideration by minimizing both the number of drones and the energy consumption assuming, at the same time, static or mobile targets. The introduction of non-linear restrictions as well as of a new more realistic energy consumption model makes the resulting optimization models (mixed integer non-linear programs) more complex than the ones presented in [Zorbas et al. \(2013\)](#). A similar problem is presented in [Di Puglia Pugliese et al. \(2015\)](#), where mobile targets are covered by a set of drones with limited energy resources. Each drone can be replaced by another drone if its energy has been depleted. The paper results show that solving to optimality the previous problem is a hard task even for small instances. The authors resort to heuristics based on the resolution of restricted mixed integer programs. The heuristics show very promising performance exhibiting a reasonable trade-off between quality of the solution and computational effort. The contributions and originality of this paper are:

- We mathematically formulate the optimal drone location problem. We provide a mathematical model to compute the optimal solution of the target coverage including 3-dimensional placement of the drones to cover all the targets.
- We provide an enhanced model that takes into account the energy of each drone.
- Based on the mathematical model, we design an optimal centralized solution to solve the static or mobile drone location problem.
- We propose two low-complexity centralized algorithms which provide scalable and efficient solutions to the drone location problem. The algorithms can solve instances of the considered problem with more than 50 targets and infinite possible positions for the drones. On the other hand, the optimal centralized algorithm can only provide solutions for up to 10 targets and 7803 possible positions for the drones.
- We propose a localized algorithm for the mobile drone location problem, where each drone autonomously cooperates with neighboring drones in order to minimize the cost. Some interesting split and merge mechanisms are incorporated in the localized algorithm whereas its performance is not far from the centralized algorithm.

The remainder of the paper is organized as follows. In [Section 2](#), we discuss the “State of the art” related to the optimal placement and drone coordination problems. [Section 3](#) introduces the optimization models, whereas [Sections 4](#) and [5](#) describe the heuristic solutions. [Section 6](#) is devoted to the presentation of the computational and simulation results collected to assess the behavior of the proposed models in terms of correctness and efficiency. The paper ends with some conclusions given in [Section 7](#).

2. State of the art

2.1. Optimal placement

The optimal placement of static or mobile devices, for

monitoring a set of targets, has already been studied in the literature from different aspects. The works closest to ours deal with the design of optimization strategies that deterministically place nodes in order to meet specific goals such as coverage and network longevity ([Kar and Banerjee, 2003](#); [Navarro-Serment et al., 2004](#); [Martínez and Bullo, 2006](#); [Dasgupta et al., 2003](#); [Wang et al., 2006](#)). In contrast to the drone location problem, where the machines can change their altitude and, thus, reduce or increase the coverage area, the targets are monitored by wireless sensor nodes with fixed monitoring range. However, the optimal node placement problem can be transformed to an optimal drone location problem, considering that the nodes can adjust their monitoring range to cover more or less targets. Other similar node placement problems are the relay node placement and placement of data collectors. A thorough review of these works is done in [Younis and Akkaya \(2008\)](#).

Other optimization problems deal with the positioning of cameras for surveillance systems ([González-Banos, 2001](#); [Bodor et al., 2007](#); [Ercan et al., 2006](#)). The purpose of the placement is to provide full coverage of a whole area as well as the highest resolution images of objects and motions in the scene that are critical for the performance of some specific task. This kind of problems differs to the drone location problem since the position of the camera is horizontal.

2.2. Similarities with sensor networks

The drone location problem with adjustable heights is similar to the node deployment problem of wireless sensor networks with adjustable sensing ranges. In this kind of networks a number of nodes are deployed to cover some targets or area while the nodes can adjust their sensing radii to conserve energy. Important information about this similar problem can be found in [Bartolini et al. \(2012\)](#), [Cardei et al. \(2005b\)](#), [Dhawan et al. \(2006\)](#), and [Zhou et al. \(2009\)](#). Most of these works deal with the problem of computing the maximum number of cover sets such that only one set is active at any time and each set covers all the targets. Adjustment of radii is used as an option of reducing the energy cost and, thus, increasing the number of sets. This type of scheduling problems are proven to be NP-Complete ([Cardei et al., 2005a](#)).

2.3. Vehicle coordination and coverage

The use of flying machines for monitoring purposes is an important task which has attracted a lot of research effort, in the recent years. We use the term “coverage” to refer to a wide range of problems related to target tracking, area discovery, area patrol, quality of detection, and navigation. On the other hand, drone coordination is usually limited to (a) how a fleet of machines can cooperatively monitor a number of ground targets as well as to (b) static or dynamic machine routing problems, related to the computation of an optimal trajectory.

A relevant big part of the literature is dedicated to coverage requirements and mainly to how well objects are tracked by sensors (e.g. cameras) attached to the drones. In [Sinha et al. \(2004\)](#) and [Gu et al. \(2006\)](#) a target detection problem is considered where a group of drones detects the position of targets using sensors located on the machines. Real vision-based drone navigation and guidance systems are presented in [Watanabe et al. \(2010\)](#) and [Brown et al. \(2006\)](#). This kind of systems is capable of localizing targets and estimates their position. In [Zhu et al. \(2013\)](#), an approach that guarantees the global convergence of a single drone to a desired orbit around a target is proposed. Constant background wind and target motion are, also, taken into account. In [Kim and Kim \(2008\)](#) and [Wang et al. \(2010\)](#) algorithms to cooperatively track a moving target by several drones are proposed.

The objective is to keep the target in the sight of the cameras from different angles and predict its motion. A patrol problem where drones must repeatedly visit a set of static targets is presented in Cannata and Sgorbissa (2011). A similar problem is presented in Chung et al. (2005), where an automated surveillance system for tracking multiple mobile ground targets is proposed. The objective of the approach is to search and cover targets while taking into account the energy restrictions of the machines.

Communication between drones can improve the location estimation of mobile targets and the robustness of the system (Ridley et al., 2003). In Schumacher (2005), a team of drones has been used to simulate a cooperative moving target engagement scenario, with the team acting as a sensor and communication network to cooperatively track and attack moving ground targets. Cooperative search and coverage is, also, the objective of Pavone et al. (2009, 2011) and Razafindralambo et al. (2010). Algorithms are proposed to divide the area into segments and machines are used to cooperatively monitor these segments.

Many works related to drones deal with the optimization of one or more objectives. The objectives may vary depending on the characteristics of the application (i.e., knowledge of the position of the targets, number of machines, full or partial coverage of the area). However, most of the scientific contributions deal with the minimization of a single criterion such as (a) the average time between the appearance of a target and the time in which drones carry out the coverage task (Hentenryck et al., 2010), (b) the total length of the trajectory (Bullo et al., 2011), and (c) the service cost (Toth and Vigo, 2001). The computation of an optimal trajectory planning is a common problem in the literature. Such problems are mathematically represented as dynamic machine routing models where the position of the targets is advertised to the machines. Solutions related to task allocation and trajectory planning are proposed in Ahmadzadeh et al. (2006), Simi et al. (2013) and Jin et al. (2003). Finally, a target tracking scenario with multiple objectives is presented in Dogan and Zengin (2006) Zengin and Dogan (2011). It is assumed that specific restricted areas must be avoided, each drone must stay within a specific distance from a target, and the total threat exposure level must be minimized.

Despite the recent extensive research effort on target tracking and drone coordination, cost efficiency is not well studied. Although some scientific works deal with the cost minimization (Qi and Zhao, 2005; Hrabar, 2008), they focus on the computation of energy efficient trajectories, where only a single or a few targets are deployed.

3. The static and mobile drone location problems

In this section, we mathematically formulate the Static and Mobile (dynamic) Drone Location Problems, referred to as *SDLP* and *MDLP* respectively. In order to describe the proposed mathematical model to represent the optimal static or mobile drone location problem, it is useful to introduce the following notations and definitions.

The flying zone is represented as a parallelepiped of height h_{max} . The target detection above h_{max} is not possible and the machines are not allowed to fly beyond this threshold. In addition, the drone cannot fly below a given height h_{min} . The projection of the whole flying region is represented by a rectangle with length x_{max} and width y_{max} .

We discretize the flying zone, thus we assume that we have points (x, y, h) where the drones could be located. It is worth observing that the targets assume arbitrary positions in the terrain of dimension $x_{max} \times y_{max}$.

Let U denote the set of available machines and T be the set of targets to be monitored. It is assumed that each target $t_i \in T$ is characterized by its coordinates (X_{t_i}, Y_{t_i}) . Given a drone u located at point (x_u, y_u, h_u) and the target t_i , we define the distance between u and t_i when $h=0$ with $D_{t_i}^{x_u y_u} = \sqrt{(X_{t_i} - x_u)^2 + (Y_{t_i} - y_u)^2}$.

Each drone u has a visibility θ that is represented by a disk in the plane (x, y) with radius r^{h_u} which depends on h_u . The higher the component h_u , the longer the radius. The visibility depends on the angle of the camera lens. Two are the main decisions to be taken. On one hand, we have to decide the position (i.e., the coordinates (x_u, y_u, h_u)) where each drone $u \in U$ should be located to monitor the targets. On the other hand, given the placement (x_u, y_u, h_u) of the drones $u \in U$, we have to decide which targets $t_i \in T$ are monitored by a drone $u \in U$. The first decision is mathematically represented by the decision variables reported below.

$$\delta_{xyh}^u = \begin{cases} 1 & \text{if the drone } u \text{ is located at the point of} \\ & \text{coordinates } (x, y, h) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The following decision variables allow us to define which targets are covered by each allocated drone:

$$\gamma_{t_i}^u = \begin{cases} 1 & \text{if the target } t_i \text{ is observed by the drone } u \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The objective is to monitor all the targets with at least one drone minimizing either the number of used machines or the total energy consumption. In addition, each drone consumes $E = (\beta + \alpha k)t + P_{max}(k/s)$ energy, where β is the minimum power needed to hover just over the ground (when altitude is almost zero) and α is a motor speed multiplier. Both β and α depend on the drone weight and the motor/propeller characteristics. P_{max} is the maximum power of the motor, s is the speed, and t is the operating time. αk denotes the relation between power and height. The term $P_{max}(k/s)$ refers to the power consumption needed to lift to height k with speed s . According to our own measurements with mobile robots (Zorbas and Razafindralambo, 2015), information by drone manufacturers and a multicopter power simulator (Müller), we presume that this power consumption model is not far from the reality. We must note here that the reader should not confuse quadcopters energy consumption with that of a fixed-wing Unmanned Aerial Vehicle whose energy consumption does not really depend on height.

The mathematical model assumes small but instantaneous drone and target movements. This assumption has been made to ease the computational process of the already overburdened optimization model. This is not exactly the case for centralized or localized heuristics where infinite number of drone and target positions is assumed. It implies that the drones movement is very small related to the previous position. Both algorithms take into consideration the maximum speed of the targets as well. This speed affects how high the drones fly as well as the time period between two algorithm computations.

The mathematical model is stated as follows:

$$\min f(\delta) \quad (3)$$

$$\text{s.t. } \sum_{(x,y,h)} \delta_{xyh}^u \leq 1 \quad \forall u \in U \quad (4)$$

$$\gamma_{t_i}^u \leq \sum_{(x,y,h)} \delta_{xyh}^u \left(\frac{r^{h_u}}{D_{t_i}^{x_u y_u}} \right) \quad \forall u \in U, t_i \in T \quad (5)$$

$$\sum_{u \in U} \gamma_{t_i}^u \geq 1 \quad \forall t_i \in T \quad (6)$$

$$\delta_{xyh}^u \in \{0, 1\}, \quad \forall (x, y, h), 1 \leq x \leq x_{\max}, \quad (7)$$

$$1 \leq y \leq y_{\max}, \quad h_{\min} \leq h \leq h_{\max}, \quad u \in U \quad (8)$$

$$\gamma_{t_i}^u \in \{0, 1\}, \quad \forall t_i \in T, u \in U \quad (9)$$

Constraints (4) ensure that the drone u is located in at most one position. Conditions (5) are used to set the value of variable $\gamma_{t_i}^u$. In particular, if the radius is less than the distance, then $\gamma_{t_i}^u$ takes a value equal to 0, otherwise, the variable can assume a value equal to either 0 or 1. Constraints (6) ensure that each target is observed by at least one drone. Constraints (7)–(9) specify the domain of the decision variables. The objective (3) is of the form $f(\delta) = \sum_{(x,y,h)} \sum_{u \in U} \delta_{xyh}^u$ when the number of used machines is minimized. On the other hand, if we want to minimize the total amount of energy consumption, $f(\delta) = \beta \sum_{(x,y,h)} \sum_{u \in U} \delta_{xyh}^u t + \alpha \sum_{(x,y,h)} \sum_{u \in U} h \delta_{xyh}^u t + \frac{P_{\max}}{s} \sum_{(x,y,h)} \sum_{u \in U} h \delta_{xyh}^u$.

The model (3)–(9) is a static representation of the problem. It is possible to formulate the problem when each target must be observed for a given amount of time. In addition, the targets can move in the field. In particular, a time window $[\tau_{\min}^{t_i}, \tau_{\max}^{t_i}]$ is associated with each target $t_i \in T$. This means that the target t_i , initially located at the point of coordinates (X_{t_i}, Y_{t_i}) , must be observed in the time range defined by the corresponding time window.

Each target moves across the area, thus it changes its position during the time. In order to capture the dynamism of the system, starting from (X_{t_i}, Y_{t_i}) , a sequence of coordinates C_i is associated with each target t_i . We assume that $|C_i| = \left\lceil \frac{\tau_{\max}^{t_i} - \tau_{\min}^{t_i}}{\Delta\tau} \right\rceil$, where $\Delta\tau$ is the time interval in which a new position of the target t_i is reached.

In order to represent the movement, target t_i is replaced with $|C_i|$ copies. A time window is associated with each copy $t_i^j, j = 1, \dots, |C_i|$ of target t_i . In particular, $[\tau_{\min}^{t_i}, \tau_{\min}^{t_i} + \Delta\tau]$ is the time window associated with the first copy of t_i , whereas the time window associated with the copy $t_i^{j+1}, j = 1, \dots, |C_i| - 1$ is defined as $[\tau_{\max}^{t_i}, \tau_{\max}^{t_i} + \Delta\tau]$. Of course, if $\tau_{\max}^{t_i} > \tau_{\max}^{t_i}$, then $\tau_{\max}^{t_i} = \tau_{\max}^{t_i}$. The copies of each target are stored in the set \hat{T} .

In this model, we have to consider other decisions. In particular, we have to state when the drones $u \in U$ both start and terminate the observation, respectively.

Let τ_{start}^u and τ_{end}^u be the initial and final time of observation of drone u , respectively.

The mathematical formulation of the optimal mobile drone location problem is reported in what follows:

$$\min f(\delta) \quad (10)$$

$$\text{s.t.} \quad \sum_{(x,y,h)} \delta_{xyh}^u \leq 1 \quad \forall u \in U \quad (11)$$

$$\gamma_{t_i^j}^u \leq \sum_{(x,y,h)} \delta_{xyh}^u \left(\frac{r_{hu}}{D_{t_i^j}} \right) \quad \forall u \in U, t_i^j \in \hat{T} \quad (12)$$

$$\sum_{u \in U} \gamma_{t_i^j}^u \geq 1 \quad \forall t_i^j \in \hat{T} \quad (13)$$

$$\tau_{\text{start}}^u \leq \tau_{\min}^{t_i^j} \gamma_{t_i^j}^u + M(1 - \gamma_{t_i^j}^u) \quad \forall u \in U, t_i^j \in \hat{T} \quad (14)$$

$$\tau_{\text{start}}^u \leq \sum_{t_i^j \in \hat{T}} \tau_{\min}^{t_i^j} \gamma_{t_i^j}^u \quad \forall u \in U, \quad (15)$$

$$\tau_{\text{end}}^u \geq \tau_{\max}^{t_i^j} \gamma_{t_i^j}^u \quad \forall u \in U, t_i^j \in \hat{T} \quad (16)$$

$$\delta_{xyh}^u \in \{0, 1\}, \quad \forall (x, y, h), 1 \leq x \leq x_{\max}, \quad (17)$$

$$1 \leq y \leq y_{\max}, \quad h_{\min} \leq h \leq h_{\max}, \quad u \in U; \quad (18)$$

$$\gamma_{t_i^j}^u \in \{0, 1\}, \quad \forall t_i^j \in \hat{T}, u \in U; \quad (19)$$

$$\tau_u, \tau_{\text{start}}^u, \tau_{\text{end}}^u \in \mathbb{R}, \quad \forall u \in U \quad (20)$$

Constraints (11) ensure that the drone u is located in at most one position. Conditions (12) are used to set the value of variable $\gamma_{t_i^j}^u$. In particular, if the radius is less than the distance, then $\gamma_{t_i^j}^u$ takes a value equal to 0, otherwise, the variable can assume a value equal to either 0 or 1. Constraints (13) ensure that each target is observed by at least one drone. Conditions (14)–(16) define the initial and final time of observation of drone u , respectively, whereas constraints (17)–(20) specify the domain of the decision variables. The value of constant M in constraints (14) is set equal to $\max_{t_i^j \in \hat{T}} \tau_{\min}^{t_i^j}$.

The solution to model (10)–(20) can be obtained by minimizing either the total number of drone by letting $f(\delta) = \sum_{(x,y,h)} \sum_{u \in U} \delta_{xyh}^u$, or the total energy consumption, that is, $f(\delta) = \sum_{u \in U} (\tau_{\text{end}}^u - \tau_{\text{start}}^u) \left(\beta \sum_{(x,y,h)} \delta_{xyh}^u \right) + \sum_{u \in U} (\tau_{\text{end}}^u - \tau_{\text{start}}^u) \left(\alpha \sum_{(x,y,h)} h \delta_{xyh}^u \right) + \frac{P_{\max}}{s} \sum_{(x,y,h)} \sum_{u \in U} h \delta_{xyh}^u$.

It is worth observing that model (10)–(20) with the objective to minimize the total energy consumption is a non-linear mixed integer program. Indeed, the objective function is non-linear.

4. Efficient solutions for SDLP

4.1. Centralized-SDLP

Due to the high complexity of the proposed optimization model, we present efficient heuristics to minimize the number of deployed machines or the total energy consumption. Centralized-SDLP (C-SDLP) solves the static drone location problem and works as a greedy heuristic. The aim of the algorithm is to minimize

either the energy consumption or the number of drones by avoiding overlappings (i.e. two or more drones covering same targets). Note that the minimization of the energy consumption does not imply the minimization of the number of drones at the same time, i.e. the two objectives are in conflict one to the other.

that the selection of the neighboring drones is done based on the distance. The closest neighboring node is examined first.

The final position of the drones is computed using a smallest enclosing circle method (SEC) (Xu et al., 2003) and the coordinates of the targets in X,Y-axes. The radius of the SEC and the angle of

Algorithm 1. Centralized-SDLP.

```

require:  $T \neq \emptyset$ , objective
1  $ACTIVE = \emptyset$ ;
  /* Phase 1: */
2 foreach  $t_i \in T$  do
3   place a drone  $u$  covering  $t_i$ ;
4    $CVRD_u = CVRD_u \cup \{t_i\}$   $ACTIVE = ACTIVE \cup \{u\}$ ;
  /* Phase 2: */
5 foreach  $u \in ACTIVE$  do
6   foreach  $u' \in ACTIVE$  do
7     if  $(dist(u, u') \leq R_c)$  then
8       set  $X, Y$  true;
9       return  $X = false$  if merge is not energy efficient (according to Formula (21) when
          objective = minimize_energy);
          return  $Y = false$  if new height exceeds  $h_{max}$ ;
10      if (1) and (2) return true then
11         $ACTIVE = ACTIVE - \{u'\}$ ;
12        foreach  $e \in CVRD_{u'}$  do
13          move targets of  $CVRD_{u'}$  in  $CVRD_u$ ;
14      compute the new position for  $u$ ;
  /* Phase 3: */
16 foreach  $u \in ACTIVE$  do
17   foreach  $u' \in ACTIVE$  do
18     foreach  $t_i \in CVRD_u$  do
19       if  $t_i$  exists in  $CVRD_{u'}$  then
20          $CVRD_u = CVRD_u - \{t_i\}$ ;
21   if  $CVRD_u = \emptyset$  then
22      $ACTIVE = ACTIVE - \{u\}$ ;
23 return  $ACTIVE$ ;

```

C-SDLP works in three phases (see Algorithm 1). During the first phase, each target is covered by a single drone with the lowest possible altitude. $CVRD_u$ is a set that keeps track of the targets u covers. In the second phase, the possibility of merging two neighboring drones is examined depending on the algorithm objectives. If the objective is to minimize the number of drones, the replacement is mandatory. If the objective is to minimize the total energy consumption, the replacement will happen only if the energy consumed by a single drone is lower than or equal to the energy of the two neighboring drones. In the second case, the following condition holds:

$$E_u + E_{u'} \geq E_{uu'} \Leftrightarrow \beta + \alpha(h_u + h_{u'} - h_{uu'})t > \frac{h_{uu'} - h_u}{s} p_{max}, \quad (21)$$

where h_u and $h_{u'}$ are the altitudes of the two drones before the merging and $h_{uu'}$ is the altitude of the single drone after the merging. After a successful merging, the drone with the lower altitude is removed, while the remaining drone checks if it can consecutively merge with another neighboring drones. The process stops when no other merging can be done. We must mention

the camera lens define the final altitude of the drone. In the last phase, drones covering already covered targets are deleted from the final group of active drones.

The longest run of Centralized-SDLP algorithm appears when all the nodes are in the neighborhood (communication range) of each other and each target is covered by a single drone. In this case, the time needed to compute the final solution would have been proportional to the time needed to examine all the pairs of neighboring drones for possible merges (lines 5–15). Every time a merge happens, the number of pairs to examine reduces by one but the cost of computing the SEC is added. However, considering that the cost of computing the SEC is low ($O(n)$, Xu et al., 2003), the worst case appears when no merges happen since the number of pair checks is maximized. It implies that the total running time is $\sum_{i=0}^{n-1} (n-i)(n-i-1)$, where $n = |T|$. We can, thus, deduce that the total running time is $O(n^3)$.

4.2. *k*-Means for SDLP

The static drone location problem can be solved by grouping

the targets into clusters and assigning a drone for each cluster. This is a fast method that has been used for many optimal node placement problems (Oyman and Ersoy, 2004). k -Means groups the targets into k clusters based on the Euclidean distance between the targets according to the clustering algorithm k -means. In this case, the problem is transformed to a problem of finding the appropriate value of k . If the objective of the deployment is to minimize the number of drones, then k must be minimized. We consecutively examine each value of k starting from 1 until all targets are covered without exceeding the maximum altitude for any of the deployed drones.

In the case of minimizing the energy consumption, all possible values of k must be examined ($1 \leq k \leq |T|$). We keep the most energy efficient layout. The position of the drones is provided by the centroid of the clusters in X -, Y -axes. The distance to the most distant member of each cluster corresponds to the radius of the circle. The radius is used to compute the altitude of each drone. We must mention here that computing the radius in this way, on one hand, the energy consumption is slightly increased since the radius is a bit longer (than that computed by a SEC method), but on the other hand, the complexity is decreased.

The complexity of the approach is based on the implementation of the k -means algorithm and the position of the points (Arthur and Vassilvitskii, 2006). If we assume that Inaba's et al. (1994) algorithm is used, then the complexity of k -means is $O(n^{2k})$, where $n = |T|$. In this case, the overall computational cost is upper bounded by the sum $\sum_{i=1}^{|T|} n^{2i}$ since $|T|$ iterations are needed with $O(n^{2k})$ complexity each ($k \in [1, \dots, |T|]$).

5. Efficient solutions for MDLP

A centralized (Centralized-MDLP) and a localized heuristic (Localized-MDLP) are presented to solve the mobile drone location problem.

In this paper, we assume that the targets have already been discovered and their initial positions have been identified. The initial placement of the drones may be done either by randomly placing them such that all the targets are covered, either by placing one drone per target, or by manually placing drones whose position has been provided by a static placement algorithm (see Section 4.1). Both centralized and localized solutions are not aware of the initial placement but their performance may be affected during the first iterations.

At each iteration, each drone covers a number of targets. The new drone positions are computed according to the movement of these targets in the meantime between two iterations. Since the movement of the targets is relatively small, the new drone position will be slightly different compared to the previous one. Note, also, that the targets never slip out of the range of a drone. This is explained in Section 5.2.

The movement of the targets is affected by multiple parameters like their mobility model (i.e., Random Way Point, Random movement, Attractors), their direction angle, and their speed. At

each instance of time, the position of the targets is updated according to these parameters. To simplify the mathematical model and reduce the computational cost, we assume discrete target positions to describe the movement of the targets. This is not the case for Localized-MDLP, which actually works with infinite number of target positions. We must, also, mention here that the future positions of the targets are not known to Localized-MDLP (neither to Centralized-MDLP). Complete knowledge is only needed for the computational model in order to obtain the optimal solution. The heuristics are only aware of the maximum speed and the initial placement of the targets.

5.1. Centralized-MDLP

Centralized-MDLP (C-MDLP) is developed to solve the dynamic drone location problem. It works in iterations and at each iteration we compute the new position of the drones. Depending on the movement of the targets (if any) the drones may slightly move towards a direction and they may split or merge with other drones. C-MDLP uses C-SDLP, described in Section 4.1, to compute the position of the drones for the first iteration. The drones' position of this first iteration is kept and updated for each of the future steps. We enhance C-SDLP to check whether two drones can be merged but, also, split to more drones according to Formula (21).

The overall complexity of the algorithm is $O(\ln^3)$, where $n = |T|$ and I is the number of iterations. The number of iterations depends on the monitoring time and how often the drones are examined for possible merges or splits. For example I coincides with the monitoring time if we assume that a merge/split check is done every second.

5.2. Localized-MDLP

In Localized-MDLP (L-MDLP) each drone acts autonomously and calculates its position based on (a) the position of the targets it currently covers, (b) the position of the targets its 1-hop neighboring drones cover, and (c) the minimum and the maximum allowed altitude. The combination of autonomy and low communication cost makes L-MDLP suitable for real monitoring applications.

We assume that the communication range is a sphere with radius R_c . Two drones that are located within the communication zone of each other can exchange messages. R_c is the same for all the nodes.

The algorithm considers that, at any time all the drones are able to estimate their position and detect the targets underneath (Krajník et al., 2011). The procedure is divided into rounds and in each round (see Algorithm 2) each drone decides its state. There are two types of state. In the *active* state, a drone monitors at least one target and communicates with other drones. In the *inactive* state, a drone abandons coverage and remains on the ground when the targets it covers can be covered by neighboring drones. This process is explained in detail in what follows.

Algorithm 2. A round of Localized-MDLP.

```

require:  $T \neq \emptyset$ ,  $ACTIVE \neq \emptyset$ ,  $objective$ 
1 foreach  $u \in ACTIVE$  do
2   detect targets of  $T$  covered by  $u$ ;
3   add detected targets in  $CVRD_u$ ;
4 foreach  $u \in ACTIVE$  do
5    $stamp_u = \rho + \frac{h_{max}}{h_u}$ ;
6   send  $stamp_u$  to 1-hop neighbors along with  $u$  position and target positions;
7 foreach  $u \in ACTIVE$  do
8   foreach  $msg$  received from neighbor  $u'$  do
9     if  $(stamp_u < stamp_{u'}) \ \&\& \ (dist(u, u') + h_u \tan \theta' + h_{u'} \tan \theta' < 2h_{max} \tan \theta')$  then
10      if  $objective = minimize\_drones$  then
11        move targets of  $CVRD_{u'}$  in  $CVRD_u$ ;
12      else if  $objective = minimize\_energy$  then
13        if  $(\beta + \alpha * (h_u - h_{u'} - h_{uu'}))\Delta t > (h_{uu'} - h_u) * \frac{P_{max}}{s}$  then
14          move targets of  $CVRD_{u'}$  in  $CVRD_u$ ;
15      foreach  $e \in CVRD_{u'}$  do
16        if  $(e \in CVRD_u) \ \&\& \ (stamp_u \geq stamp_{u'}) \ \&\& \ (dist(u, u') < (h_u \tan \theta' + h_{u'} \tan \theta'))$  then
17           $CVRD_u = CVRD_u - \{e\}$ ;
18 foreach  $u \in ACTIVE$  do
19   if  $CVRD_u \neq \emptyset$  then
20      $CALLS_u = \emptyset$ ;
21     if  $objective = minimize\_energy$  then
22        $CVRD_u^{temp} = CVRD_u$ ;
23        $CVRD_u^{temp} = CVRD_u^{temp} - \{most\_distant\_target\}$ ;
24       compute a temporary altitude  $h_u^{temp}$  using  $CVRD_u^{temp}$ ;
25       if  $(\beta + \alpha * (h_u - h_{u'} - h_{uu'}))\Delta t > (h_{uu'} - h_u) * \frac{P_{max}}{s}$  then
26          $CALLS_u = CALLS_u \cup \{most\_distant\_target\}$ ;
27          $h_u = h_u^{temp}$ ;
28          $CVRD_u = CVRD_u^{temp}$ ;
29       compute new altitude  $h'_u$ ;
30       while  $h' > h_{max}$  do
31         find the most distant target in  $CVRD_u$ ;
32          $CVRD_u = CVRD_u - \{most\_distant\_target\}$ ;
33          $CALLS_u = CALLS_u \cup \{most\_distant\_target\}$ ;
34         compute new altitude  $h'_u$ ;
35       call one or more drones to cover the targets in  $CALLS_u$ ;
36       update position;
37   else
38      $ACTIVE = ACTIVE - \{u\}$ ;

```

Each round starts by detecting the targets, that each active drone u covers. u maintains the information about the targets it covers in $CVRD_u$. The initial number of active drones depends on the initial placement. As explained before, the initial placement of the robots can be done either randomly (covering all the targets) or by placing them manually since the initial position of the targets is known. Once the targets have been detected and their position has been identified, each active drone computes a stamp taking into account its current altitude and a random value ρ . ρ is used to avoid having two drones with same stamp.

The stamp, as well as the coverage info of each active drone, is broadcasted to 1-hop neighbors and the received stamps as well as the coverage info of the neighboring drones are used to decide if a

drone will discard certain targets while computing its new location or not. During this decision, merges or splits may happen.

A *merge* takes place when two neighboring drones are very close to each other and the targets they cover together can now be covered using only one of the drones. The process is similar to that described in [Section 4.1](#). That is: if the objective is to minimize the energy cost, two neighboring drones may merge to one if the energy expense after the merging is lower than having two active drones. On the other hand, two drones always merge to one if the objective of the algorithm is to minimize the number of drones.

The feasibility of a merging is examined using two formulas. Formula (22) makes sure that the remaining drone will not exceed the highest possible altitude:

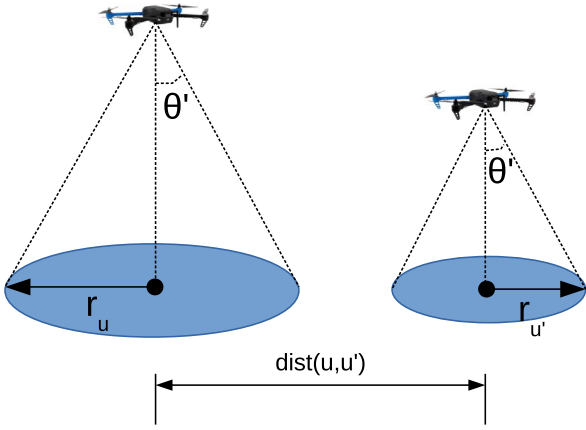


Fig. 1. Drone u and drone u' and the possibility of merging.

$$\frac{dist(u, u') + h_u \tan \theta' + h_{u'} \tan \theta'}{2 \tan \theta'} < h_{max}. \quad (22)$$

$dist(u, u')$ represents the distance between the centers of the two disks and $dist(u, u') + h_u \tan \theta' + h_{u'} \tan \theta'$ is the maximum possible diameter of the merged coverage disk (see Fig. (1)).

In order to manage merges, L-SDLP uses a slightly different formula from that presented in Section 4.1:

$$E_u + E_{u'} \geq E_{uu'} \Leftrightarrow (\beta + \alpha(h_u + h_{u'} - h_{uu'}))\Delta t > \frac{h_{uu'} - h_u}{s} p_{max}. \quad (23)$$

Here, Δt denotes the time period during which the merged drone will remain at an energy efficient height and no split will happen. This period of time clearly depends on the behavior of the targets. If the covered targets are moving towards opposite directions, this period of time must be short since a split would most likely happen in a few iterations.

In case of a merging the targets covered by the drone with the highest stamp are excluded from its $CVRD_u$. Moreover, targets that enter into the range of a drone with higher altitude are, also, excluded from this set. The drone with the lowest altitude performs the opposite process; it considers the targets of the neighboring drones as members of its $CVRD_u$.

On the other hand, a *split* may happen only if the algorithm objective is the minimization of the energy consumption or if a drone is going to reach the maximum possible altitude. Each drone u uses the same condition as in Centralized-SDLP, described by Formula (21), to check if it is more efficient to split into two drones or not. More specifically, u checks if the total energy consumption can be decreased whilst discarding the most distant event and calling a new drone to cover it. In the affirmative case, the most distant event is added in set $CALLS_u$ and it is removed from $CVRD_u$.

Every time a split occurs, a new drone which is located at the ground moves to cover the uncovered targets. The coordinates are computed by the drone which decides to split and they are communicated to the closest command center. In this paper, we assume square terrain sizes whose side does not exceed 100 m. Assuming that the command center is located at the middle of the terrain, the maximum drone traveling distance is $\sqrt{(\sqrt{2}50)^2 + h_{max}^2}$. Modern drones can move with speeds up to 50 km/h (or 13.9 m/s). It means that the longest responsiveness of the drone is less than 5.5 s. During this period of time, the targets cannot slip out of the drone range unless the maximum target speed is higher than 1 m/s. For larger terrain sizes or higher target speeds, multiple command centers should be used. Responsiveness is assessed in Section 6.2.2.

Since the feasibility of merges and splits has been examined and changes in covering sets have been made, each drone u

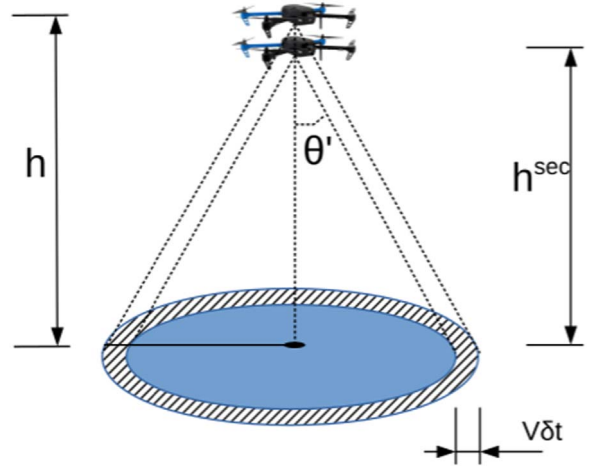


Fig. 2. Altitude difference between that computed by SEC and the actual one.

continues with the computation of its final position. The targets in $CVRD_u$ are used by an enclosing circle subroutine (or any other method) to determine the final drone position. Once the new position does not exceed the maximum allowed altitude, the drone is moving to the new position and a new round of L-MDLP begins.

If the final altitude has exceeded the maximum allowed altitude, the algorithm discards the most distant target until the new altitude is below this value. The discarded targets are moved to set $CALLS_u$. One or more drones are called to cover the targets in $CALLS_u$ using a broadcast message to the closest command base. The message contains information about the position of the targets and the final position of the new drone.

In the meantime between two rounds, one or more targets cannot slip out of the coverage range. Depending on the speed of the targets and how often the drones detect the targets, the actual altitude of the drones must be slightly higher than that computed by the SEC algorithm (or any other method) (see Fig. 2). The final drone altitude is computed using the following formula:

$$h_u = h_u^{SEC} + V I_t \tan \theta', \quad (24)$$

where V is the horizontal speed of the drones, I_t is the time period between two target detections and h_u^{SEC} is the altitude calculated by the SEC algorithm (or by any other suboptimal solution). $V I_t$ is, in fact, the maximum distance that a target can travel and it has been added to ensure that these targets will be still covered until the next detection. Simulation results with different values of I_t are presented in Section 6.2.3.

In scenarios where a drone may cover several targets, the computation of the smallest enclosing circle may be slow. A possible delay in the computation may cause abnormal behaviors, such as the appearance of uncovered targets (some targets may slip out of the range of the drone due to a delay of the computation of the altitude of the previous iteration). For that reason suboptimal solutions with lower computation cost can be used (e.g. the average value of the coordinates of the points).

Concerning the message cost, L-MDLP uses two types of messages to communicate with other drones. The first type is a broadcast packet containing information about the stamp, drone position and the targets it monitors. The second type is used to call other drones when discarded targets have to be covered. A possible packet loss of the first message type does not affect the monitoring of the targets since, in the worst case, it could only cause a double covering of some of the targets. On the other hand, a packet loss of the second message type leads to uncovered targets if the discarded target is not covered by any other drone at

that particular moment. We assess the effect of packet loss on the number of uncovered targets in [Section 6.2.1](#).

The communication complexity of the L-MDLP depends on the number of active drones and their position in the space. The maximum number of messages is sent when the number of drones is maximized; that is when each target is covered by a single drone and no merges are made throughout the monitoring time. Since all the nodes send a single message at the beginning of each round, the maximum communication complexity is nI .

L-MDLP is based on LAS algorithm ([Zorbas et al., 2013](#)). However, it differs from LAS since it is designed to minimize both the energy consumption and the number of drones. LAS is designed to minimize the energy consumption only. Additionally, in L-MDLP the performance is enhanced (a) by allowing a drone to discard some targets in order to save energy (split process) and (b) by calling the least possible number of drones in case of a split. LAS considers no splits except if a drone exceeds the maximum altitude and every time there is a split it calls a number of drones equal to the discarded targets (higher cost).

6. Simulation and computational results

The model (3)–(9), aimed at minimizing the number of used drones and the energy consumption, and model (10)–(20) aimed at minimizing the number of drones have been implemented in Java language and solved by CPLEX. The dynamic model with the minimization of the energy consumption has been implemented in GAMS 23.6 (<http://www.gams.com>) and solved with the BONMIN solver. We set BONMIN to use a branch-and-bound algorithm which is a hybrid of B-BB and B-QG and is based on solving either a continuous non-linear or a continuous linear program at each node of the search tree, improving the linear program by outer approximations, and branching on integer variables. Since the higher complexity of the model, no optimal solution is found for all the considered instances (described in the next section). For this reason, we present only results obtained with the linear models, that is, models (3)–(9) with the minimization of the number of drones and total energy consumption and model (10)–(20) that minimizes the total number of used drones. The computational results have been carried out on an Intel(R) Core(TM) i7 CPU M620, 2.67 GHz, 4 GB machine under Microsoft 7 operating system.

On the other hand, in order to conduct the simulations, we developed a custom simulator written in Perl. The simulations have been carried out on an Intel Core2 Duo 1.67 GHz CPU under Debian/GNU Linux operating system. The usage of RAM was minimal and no parallel processing was considered.

We consider a set of 600 scenarios by letting different values for $|T|$, and number of coordinates (x, y, h) , called in the sequel $\#c$. This value indicates the number of possible points in which each drone can be placed. In addition, for each couple of values $|T|$ and $\#c$, we have generated 20 instances in which different positions of the targets are considered. In this respect, the values of X_{t_i} and Y_{t_i} , $\forall t_i \in T$, are chosen randomly in the range $[0, 100]$. In addition, we imposed a time limit of 1 h for the execution time to solve the models.

In particular, we set $|T| = 10, 15, 20, 25, 30, 50$, $\#c = 108, 363, 1323, 7803, 30, 603$, $x_{max}, y_{max} = 100$ m, $\alpha = 10.5$, $\beta = 30$, $P_{max} = 85$, $R_c = 50$ m, and $h_{max} = 10$. Three different values for the altitude h have been considered, i.e., $h = 1, 5, 10$ m. In order to test the dynamic model, we have considered the static instances, described above, in which time windows of observation have been introduced. We set the width of the time windows to 10 for each target. It is worth observing that the dynamic instances

are equal to the static counterparts. Indeed, they are characterized by the same number of targets placed in the same position. To model the dynamism, we assume that targets t_i and t_{i+1} are the copies of the same target. In other words, $|T| = |T|$. A camera with wide lens of 120° was considered for all the simulation results. The selection of this value is crucial for the computation cost since it is highly related to $\#c$ values. If the angle is small, high values of $\#c$ must be selected, increasing a lot of computational complexity of the model.

The simulation results are divided into two parts. In the first part, we compare the simulation results to the ones obtained by solving the optimization models. For this reason, the same limited values regarding $\#c$, h and time window were used. In the second part, we consider infinite values for $\#c$ and h ($1 \leq h \leq 10$). We, also, set a time window equal to one and we allow each event to move up to 500 times. We keep the same values for the rest of the parameters. For the localized algorithm, we assume that the targets are detected every one round unless a different value is specified in the text. The vertical speed of the drones is 2 m/s and their horizontal 40 km/h (11.11 m/s). Moreover, regarding the mobility of the targets, we assume a random mobility model where the targets initially choose a direction and in each step the position is updated according to a maximum angle and a maximum speed. The maximum angle and speed are equal to $\pi/4$ and 1 m/s, respectively. Other mobility models and target distributions such as the *Random Way Point* and the *Attractors* have been evaluated as well, but the random mobility model was selected as the most generic one. For more information about the assessment of the mobility models you can check <http://uav-scheduling.gforge.inria.fr/>.

Since the positions of the targets are chosen randomly during the process, each of the 20 generated instances is executed 10 times and the average costs are presented (i.e., 200 executions per scenario). The 95% confidence intervals are, also, shown in the figures when relevant. The initial placement of the targets is provided by C-SDLP in order to have a fair comparison with the computational results. Other placement options have been, also, tested exhibiting a slightly higher cost (number of drones or total energy) during the first iterations of the algorithm.

6.1. Results for scenarios with limited input

In this section, we discuss the results obtained using limited values for $\#c$, h and time window. The results are summarized in four tables ([Tables 1–4](#)) collected on the generated scenarios when solving model (3)–(9) and (10)–(20), respectively. Each row reports the average results over the instances. Concerning the optimization models, we include both optimal solutions and feasible solutions given by the solver within the imposed time limit. We report the value of the objective function under column *cost*, the execution time in seconds under column *time*, the number of solved scenarios to optimality under column *opt*, the number of scenario for which a feasible solution is available under column *#feas*, and the number of scenario for which the solver is not able to produce a feasible solution within the time limit is reported under column *#!feas*. In SDLP, we compare the performance of the two centralized algorithms (i.e., C-SDLP and *k*-means) with the solutions of the proposed optimization models. In the dynamic model, we assess the performance of C-MDLP and L-MDLP and we compare the related results to those obtained by solving the optimization models.

The numerical results underline that the problem of minimizing the number of drones is easier than that for which the total energy consumption is minimized (see [Tables 1 and 2](#)). Indeed, the solver is able to solve to optimality 448 and 265 instances when the number of drones and the energy consumption is minimized,

Table 1

Average simulation and computational results of the static drone location problem while minimizing the number of drones.

N	#c	Minimize number of drones								
		Computational					C-SDLP		k-Means	
		Cost	Time	#opt	#feas	#!feas	Cost	Time	Cost	Time
10	108	6.15	0.01	20			6.38	0.0014	6.94	0.0013
10	363	5.50	0.04	20			5.68	0.0014	5.89	0.0010
10	1323	5.00	1.82	20			5.19	0.0014	5.63	0.0010
10	7803	4.75	168.93	19	1		4.98	0.0014	5.29	0.0009
10	30,603	4.75	276.74	19	1		4.92	0.0014	5.46	0.0009
15	108	8.10	0.02	20			8.63	0.0029	10.49	0.0034
15	363	6.40	0.11	20			6.87	0.0027	7.56	0.0021
15	1323	6.05	468.18	18	2		6.51	0.0027	7.01	0.0019
15	7803	5.95	347.69	18	2		6.34	0.0030	6.95	0.0019
15	30,603	5.75	1190.77	13	7		6.17	0.0030	6.85	0.0018
20	108	9.70	0.04	20			10.36	0.0049	12.17	0.0057
20	363	7.60	0.26	20			8.21	0.0042	8.99	0.0035
20	1323	6.85	1183.23	12	8		7.52	0.0044	8.55	0.0033
20	7803	6.55	1138.99	12	8		7.18	0.0050	8.30	0.0032
20	30,603	6.40	989.65	15	5		7.03	0.0050	8.23	0.0032
25	108	10.50	0.06	20			11.44	0.0073	14.46	0.0088
25	363	8.15	0.40	20			9.37	0.0068	10.41	0.0057
25	1323	7.45	1275.10	9	11		8.50	0.0068	9.62	0.0051
25	7803	7.20	750.27	15	5		8.02	0.0077	9.48	0.0050
25	30,603	7.26	929.79	12	7	1	7.92	0.0078	9.64	0.0051
30	108	11.50	0.10	20			12.63	0.0109	18.35	0.0149
30	363	8.90	0.64	20			10.07	0.0104	12.19	0.0089
30	1323	7.90	748.04	14	6		9.04	0.0092	11.09	0.0078
30	7803	7.79	1728.95	7	12	1	8.70	0.0111	10.41	0.0072
30	30,603					20	8.57	0.0112	10.27	0.0070
50	108	14.80	0.33	20			15.74	0.0308	27.06	0.0516
50	363	10.85	2.20	20			12.25	0.0300	16.59	0.0269
50	1323	9.50	2579.38	3	17		10.85	0.0273	14.77	0.0229
50	7803	9.85	2283.64	2	18		10.50	0.0346	14.28	0.0218
50	30,603					20	10.46	0.0336	14.36	0.0220

Table 2

Average simulation and computational results of the static drone location problem while minimizing the energy consumption.

N	#c	Minimize energy consumption								
		Computational					C-SDLP		k-Means	
		Cost	Ttime	#opt	#feas	#!feas	Cost	Time	Cost	Time
10	108	408,163.25	0.01	20			420,389.94	0.0027	466,583.75	0.0039
10	363	342,663.63	0.04	20			349,754.47	0.0028	358,986.36	0.0039
10	1323	308,903.13	778.51	17	3		317,337.25	0.0032	331,133.78	0.0039
10	7803	225,869.63	2489.37	4	16		239,773.62	0.0049	236,771.54	0.0039
10	30,603	222,269.63	2380.48	4	16		237,339.38	0.0058	234,665.76	0.0039
15	108	551,034.38	0.03	20			574,215	0.0058	683,436.5	0.0092
15	363	434,031.13	0.11	20			456,261.96	0.0056	488,489.15	0.0093
15	1323	386,643.13	3600.00		20		406,207.64	0.006	434,307.9	0.0093
15	7803	321,978.13	3600.00		20		365,137.5	0.0131	352,390.89	0.0093
15	30,603	320,812.38	3600.00		20		363,920.38	0.0149	348,375.96	0.0093
20	108	661,018.38	0.05	20			701,237.3	0.0095	815,119.46	0.0179
20	363	519,321.88	0.24	20			550,501.15	0.0088	588,265.79	0.0179
20	1323	459,342.75	3600.00		20		490,021.91	0.0105	532,438.34	0.0179
20	7803	396,529.13	3600.00		20		479,547.25	0.0263	450,994.44	0.0179
20	30,603	399,477.13	3600.00		20		475,895.88	0.0293	451,354.86	0.0179
25	108	741,226.25	0.07	20			794,224.8	0.0138	984,533.62	0.0309
25	363	570,233.75	0.41	20			634,826.7	0.0139	687,425.55	0.031
25	1323	502,806.63	3600.00		20		552,667.79	0.0163	611,811.41	0.0309
25	7803	445,169.75	3600.00		20		596,391.25	0.0595	544,682.04	0.0377
25	30,603	476,735.13	3600.00		20		586,654.25	0.0534	538,131.75	0.0309
30	108	812,288.88	0.12	20			870,964.15	0.0239	1,248,198.65	0.0554
30	363	635,271.00	0.65	20			699,507.12	0.0215	795,181.18	0.0485
30	1323	551,121.25	3600.00		20		610,349.51	0.0233	704,477.54	0.0486
30	7803	512,708.50	3600.00		20		712,018.12	0.0817	631,194.86	0.0486
30	30,603					20	701,064	0.1005	624,116.06	0.0565
50	108	1,054,730.25	0.41	20			1,111,059.31	0.0585	1,854,050.86	0.1876
50	363	807,480.63	2.42	20			891,676.15	0.064	1,135,196.64	0.1883
50	1323	694,617.75	3600.00		20		780,524.03	0.0738	994,622.82	0.2103
50	7803	741,191.05	2817.36		19	1	1,117,320.75	0.4037	924,309.16	0.1986
50	30,603					20	1,084,701.8	0.397	916,712.18	0.1969

Table 3

Average simulation and computational results of the mobile (dynamic) drone location problem while minimizing the number of drones.

T	#c	Minimize number of drones								
		Computational					C-MDLP		L-MDLP	
		Cost	Time	#opt	#feas	#!feas	Cost	Time	Cost	Time
10	108	6.15	0.01	20			6.45	0.0337	7.23	
10	363	5.50	0.04	20			5.61	0.0307	6.44	
10	1323	5.00	1.82	20			5.24	0.0324	6.2	
10	7803	4.75	171.33	19	1		4.96	0.0302	5.88	
10	30,603	4.75	272.24	19	1		5	0.034	5.77	
15	108	8.10	0.02	20			8.31	0.0594	9.54	
15	363	6.40	0.11	20			6.81	0.0515	8.19	
15	1323	6.05	467.51	18	2		6.4	0.0545	7.74	
15	7803	5.95	350.98	18	2		6.19	0.0557	7.28	
15	30,603	5.75	1202.55	13	7		6.19	0.06	7.27	
20	108	9.70	0.04	20			9.86	0.0882	11.51	
20	363	7.60	0.23	20			8.12	0.079	9.91	
20	1323	6.85	1308.07	12	8		7.4	0.0815	9.32	
20	7803	6.55	1216.64	12	8		7.11	0.0846	8.95	
20	30,603	6.40	1029.03	15	5		7.15	0.0908	8.73	
25	108	10.50	0.06	20			10.98	0.1327	13.18	
25	363	8.15	0.40	20			8.94	0.1099	11.05	
25	1323	7.45	1781.31	9	11		8.19	0.1142	10.48	
25	7803	7.20	811.19	15	5		7.9	0.1186	10.01	
25	30,603	7.42	984.59	11	8		7.8	0.1268	9.82	
30	108	11.50	0.10	20			12.2	0.1689	14.99	
30	363	8.90	0.63	20			9.6	0.1457	12.36	
30	1323	7.90	936.91	14	6		8.83	0.15	11.58	
30	7803	7.80	1885.74	8	12		8.44	0.1591	11.01	
30	30,603					20	8.39	0.1667	10.85	
50	108	14.84	0.32	20			15.55	0.1234	20.21	
50	363	10.85	2.19	20			12.08	0.1125	16.33	
50	1323	9.50	2735.57	3	17		10.93	0.1124	15.17	
50	7803	9.67	2006.19	2	16	2	10.34	0.1224	14.24	
50	30,603					20	10.28	0.1271	14.13	

Table 4

Average simulation and computational results of the mobile (dynamic) drone location problem while minimizing the energy consumption.

T	#c	Minimize energy consumption								
		Computational					C-MDLP		L-MDLP	
		Cost	Time	#opt	#feas	#!feas	Cost	Time	Cost	Time
10	108						17,783.89	0.0349		21,608.48
10	363						15,852.54	0.0533		16,428.49
10	1323						14,280.53	0.0534		15,918.1
10	7803						8572.9	0.0743		17,087.2
10	30,603						8434.76	0.0868		9234.9
15	108						23,576.08	0.0641		29,001.82
15	363						22,189.69	0.1131		23,378.44
15	1323						19,426.44	0.1029		21,522.75
15	7803						12,812.73	0.1982		23,954.75
15	30,603						12,602.04	0.2073		13,950.1
20	108						28,456.35	0.0958		36,757.47
20	363						27,296.52	0.194		28,441.26
20	1323						23,970.31	0.1664		26,136.9
20	7803						17,019.99	0.3841		29,462.55
20	30,603						16,705	0.4299		18,654.8
25	108						32,133.69	0.1312		43,436.46
25	363						31,261.31	0.2483		33,120.73
25	1323						27,375.86	0.2555		30,144.36
25	7803						21,139.42	0.6948		33,716.49
25	30,603						20,695.96	0.7239		22,632.7
30	108						35,813.51	0.1784		50,381.56
30	363						36,033.97	0.3873		37,929.85
30	1323						30,992.6	0.3381		34,536.03
30	7803						25,288.86	1.0501		39,223.2
30	30,603						24,721.37	1.121		26,804.91
50	108						47,263.94	0.1336		74,204.65
50	363						51,256.95	0.3201		54,379.24
50	1323						41,672.91	0.23		48,295.74
50	7803						41,680.63	1.4265		55,234.27
50	30,603						40,224.37	1.4017		40,613.72

respectively. In addition, the computational cost for solving the latter is 3.22 times higher than that required for the former.

Comparing the computational results collected with static and dynamic models, an interesting behavior can be observed.

First, the number of scenarios solved to optimality under the dynamic setting is equal to that obtained with the static counterpart. However, the number of dynamic scenarios for which the solver is not able to produce a feasible solution within the time limit is equal to 43, whilst the solver fails to provide a feasible solution for 42 scenarios in the case of static problem. The execution time for solving the static problem is closer to the time required by the dynamic counterpart. Indeed, the latter is 1.06 slower than the former. One expects that the dynamic problem should be harder than the static one. The obtained results can be justified by considering that the introduction of constraints (14) and (16) reduces the feasible region and this reduction suffices the higher complexity due to the introduction of variables τ_{start}^u and τ_{end}^u . It follows that the search process for the dynamic model is similar to the static one.

Secondly, the average optimal solution value of the dynamic model is very close to the static model. This is an expected trend since the number of targets is the same for the two groups of instances (i.e., static and dynamic).

Third, Tables 1–4 show that the higher the $\#c$, the more difficult is the scenario. This behavior is justified by considering the number of variables of the model. Indeed, the higher the $\#c$, the higher the number of possible positions that each machine can occupy. Of course, the lower the $\#c$, the smaller the feasible region.

This aspect justifies the value of the objective function. Indeed, the higher the $\#c$, the better the optimal solution value. This behavior can be observed for both cases, i.e., minimization of number of drones and energy consumption, respectively.

The simulation results of Tables 1 and 2 show that the performance of C-SDLP and k -means is very close to the best solution provided by solving the optimization model. C-SDLP presents better results than k -means specially in the case where many targets are deployed. On the other hand, k -means is faster for the majority of the scenarios. Even if the optimization model and the simulations were executed on different machines, we can observe that the two heuristics present an almost linear trend in the execution time compared to the computational cost required by solving the optimization model. Apparently, due to their lower complexity, the heuristics are able to solve all the instances in very small amount of time.

Concerning the dynamic model, the results presented in Tables 3 and 4 show that the localized solution performs up to 20% worse than the centralized one (except when $\#c = 7803$). However, L-MDLP's performance is quite acceptable since in L-MDLP (a) the decision of merges and splits is done locally and no global optimization is performed, and (b) the altitude of the drones is computed using the average coordinates of the targets which results to a higher energy consumption.

Figs. 3 and 4 depict a representative example of the positions of the drones as they have been computed by the three solutions when the number of drones and the energy consumption is minimized, respectively. An instance of the static problem with 15

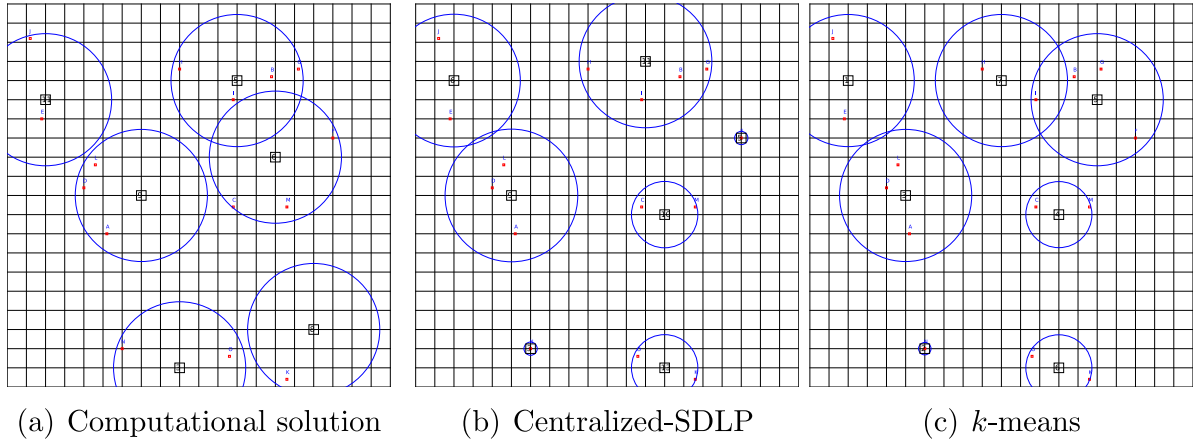


Fig. 3. The positions of the drones computed by the three solutions when 15 targets are deployed, $\#c = 1323$, and the number of drones is minimized.

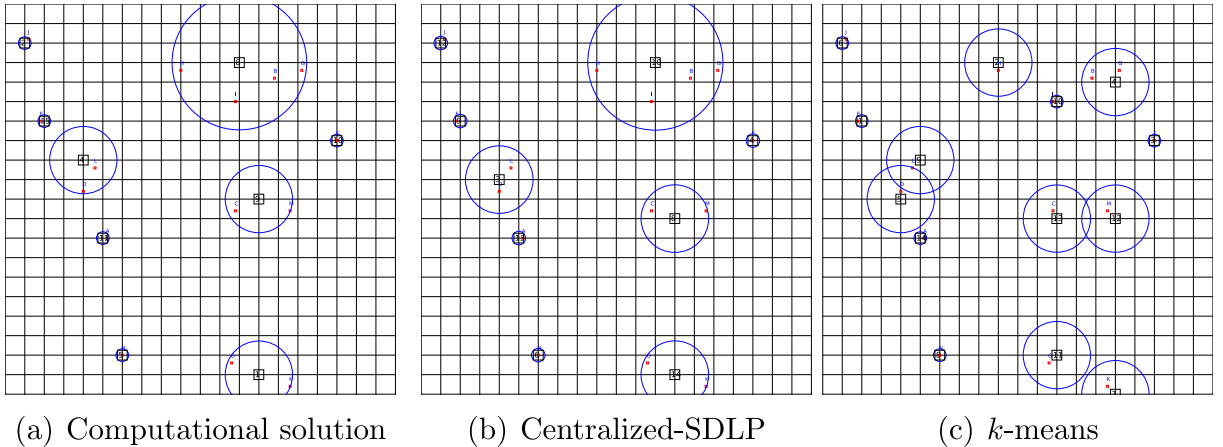


Fig. 4. The positions of the drones computed by the three solutions when 15 targets are deployed, $\#c = 1323$, and the energy consumption is minimized.

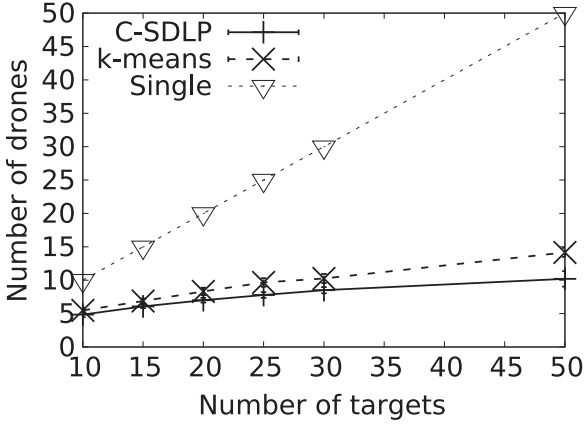


Fig. 5. Minimization of the number of drones for the static drone location model.

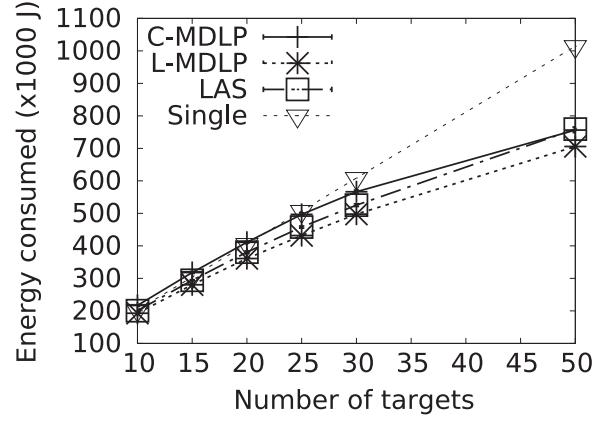


Fig. 8. Minimization of the energy consumed for the dynamic drone location model.

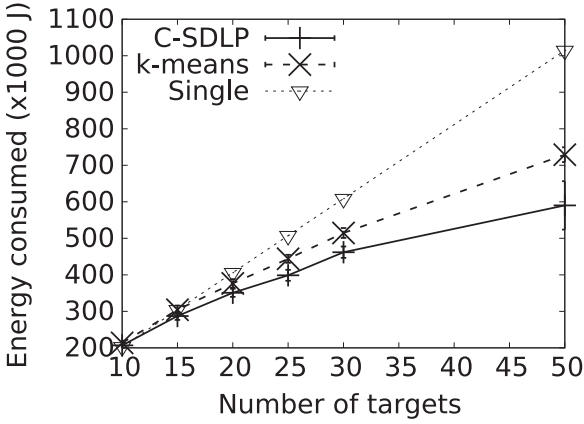


Fig. 6. Minimization of the energy consumed for the static drone location model.

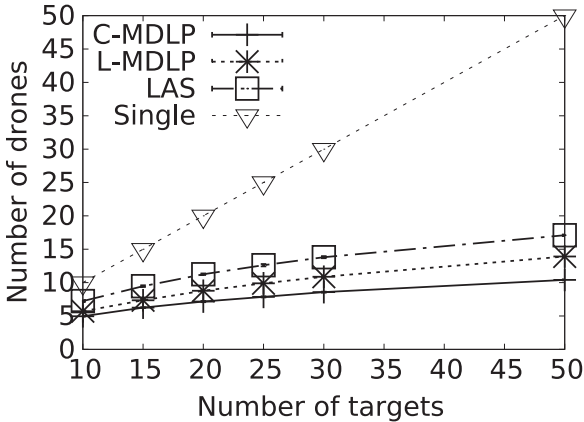


Fig. 7. Minimization of the number of drones for the dynamic drone location model.

targets and grid tics of 5 m ($\#c = 1323$) were used. The best solution is provided by the computational model with 6 drones/293,775 Joules, while C-SDLP and k -means follow with 7 drones/293,775 Joules and 7 drones/453,455 Joules, respectively.

6.2. Results for scenarios with unlimited input

In this section, we assess the centralized and localized algorithms in more complex scenarios without restrictions in the

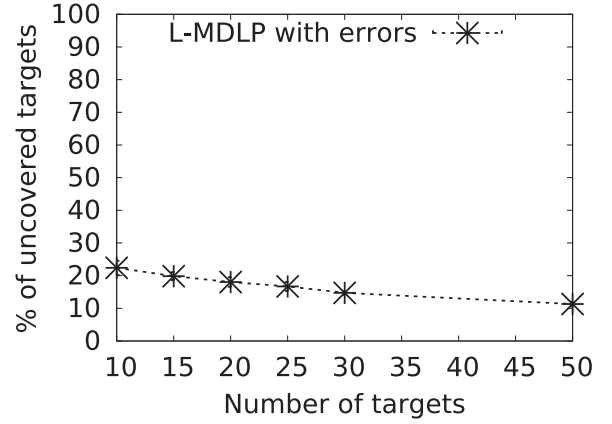


Fig. 9. Percentage of uncovered targets when the number of drones is minimized and communication errors occur.

number of available altitudes, positions, time windows and number of rounds. We compare their performance to that of “Single” deployment. We call “Single” the deployment where each target is covered by a single drone and it denotes the worst case deployment in terms of cost.¹

Figs. 5 and 6 confirm the behavior we observed in the previous section, showing that C-SDLP performs slightly better than k -means, when the number of drones or the energy is minimized, respectively. Both algorithms are far away from the worst case scenario of “Single” deployment when the objective is the minimization of the drones. When the objective changes, the gap is smaller since for many scenarios the best solution coincides with the worst case.

The rest of the results are related to the mobile (dynamic) drone location problem. The centralized and localized solutions are compared to LAS algorithm (Zorbas et al., 2013) and to “Single”. Fig. 7 depicts the performance of the four approaches, when the number of drones is minimized. As it was expected the centralized algorithm presents the best performance, however, L-MDLP is very close to the centralized one while it outperforms LAS. When the objective is to minimize the energy consumption (see Fig. 8), all the approaches perform very close to “Single” for low target populations since a few only merges can happen. The gap is bigger as the number of targets increases.

¹ When the objective is the minimization of the number of drones, the worst case deployment is upper bounded by the number of drones needed to cover the whole terrain area. In our case, where a 100×100 m terrain is used, this number is equal to 25.

6.2.1. Presence of communication errors

L-MDLP's tolerance in communication errors is assessed in the next set of simulations (see Figs. 9 and 10). We assume that there is a 50% probability of losing a message when it is transmitted to a neighboring drone or a command center, while we measure the percentage of uncovered targets. When a message is lost, one or more targets may be left uncovered as it has been explained in Section 5.2. Despite the huge loss probability, the percentage of uncovered targets throughout the monitoring time is low and ranges from 11% to 22%. The number of uncovered targets is slightly higher when the objective is the minimization of the drones and few targets are placed in the terrain. This happens because less drones are deployed in the field and each drone monitors on average many targets. In this case, the maximum possible altitude is reached more easily and considering that a new drone may not arrive to cover the discarded targets, the appearance of uncovered targets is more probable. On the other hand, when many targets are placed, an uncovered target will be most likely covered by a neighboring drone due to the higher drone density. When we minimize the energy consumption we have about the same amount of uncovered targets since more drones are deployed (higher density) but they stay at lower heights (smaller area covered). It means that less splits can happen (thus less uncovered targets), but the targets will probably stay uncovered by longer period of time due to the lower height of the drones.

6.2.2. Responsiveness

The responsiveness of the new drones to splits is presented in Figs. 11 and 12. As explained in Section 5.2, we assume that in case

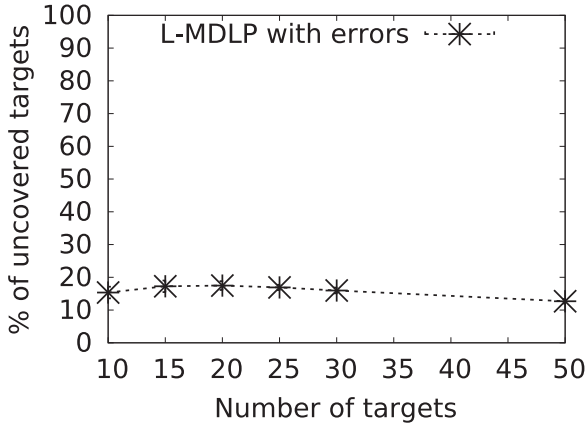


Fig. 10. Percentage of uncovered targets when the energy consumption is minimized and communication errors occur.

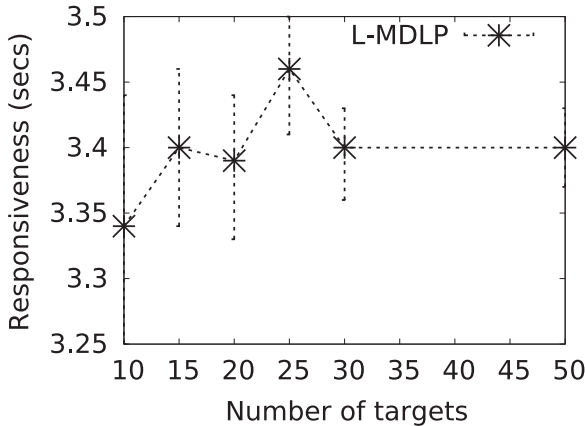


Fig. 11. Responsiveness when the number of drones is minimized.

of a split, new drones arrive from a near command station to cover the discarded targets. The time needed to cover these targets is called "responsiveness" and it depends on the position of the targets. We can observe that responsiveness ranges between 3.25 and 3.7 s and it, actually, means that no target can slip out of the range of a new drone within this low period of time.

6.2.3. Detection frequency

In the final set of simulations, we vary the detection frequency (i.e., the time elapsed between two successive target detections) and we measure the number of drones and the total energy

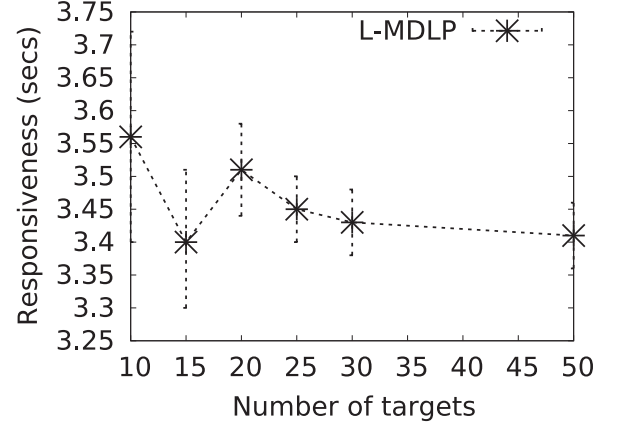


Fig. 12. Responsiveness when the total energy consumption is minimized.

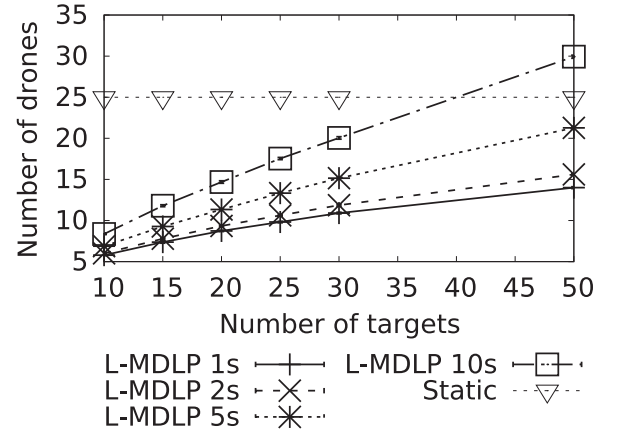


Fig. 13. Number of drones used for different detection frequencies.

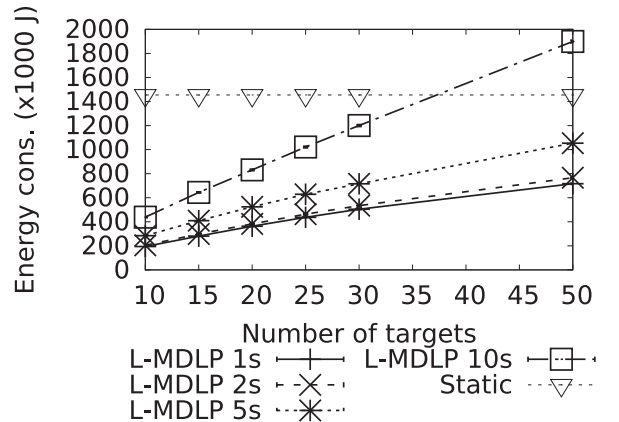


Fig. 14. Total energy consumption for different detection frequencies.

consumption. According to Formula (24), the final altitude of a drone depends on the detection frequency, which practically means that the drone should fly higher in order to still be able to detect the mobile targets when they move. Here, we compare L-MDLP against a static deployment where a minimum number of drones is used to monitor the whole area. Four detection frequencies are examined (i.e., 1, 2, 5, and 10 rounds). The results are presented in Figs. 13 and 14 and show that for high detection frequencies (e.g. 10 s) and many targets, the number of drones and the total energy consumption are slightly higher than covering the whole area with static nodes.

7. Conclusion and future work

In this paper we dealt with the cost minimization problem related to the optimal placement of drones in order to cover a set of static or mobile targets. We distinguished two minimization problems: the minimization of the number of drones and the minimization of the total energy consumption (related to drone position). We formulated the problems above and we provided mathematical models to compute the optimal solutions. Due to the high complexity of the models, we also proposed efficient heuristics as well as a practical localized solution in which each drone cooperates with other machines in its vicinity to minimize the coverage cost. The evaluation of the proposed centralized heuristics and localized solutions using simulations and solving the mathematical models showed that the centralized algorithms perform similar to the optimal solutions while the localized approach is 10–20% close to the optimal solution.

The next step of this work will focus on the detection quality. We will add another dimension and constraint to our problem by defining a threshold quality at which each target should be observed. This detection quality will be linked to the altitude of the drone and the number of drones observing a target. We will, also, introduce a mobility pattern for each target that could be learned by the drones in order to anticipate the movements of the targets and to increase the efficiency of tracking. In a long term view, the work in this paper could also include heterogeneous devices like static sensors, cars or user mobile phones. The drone tracking system will then take advantage of these device to optimize their observations.

References

- Ahmadzadeh A., Buchman G., Cheng P., Jadbabaie A., Keller J., Kumar V., Pappas G., 2006. Cooperative control of uavs for search and coverage. In: AUVSI Unmanned Systems Conference.
- Arthur D., Vassilvitskii S., 2006. How slow is the k-means method? In: Proceedings of the Twenty-second Annual Symposium on Computational Geometry, SCG '06. ACM, New York, NY, USA, pp. 144–153.
- Bartolini, N., Calamoneri, T., La Porta, T., Petrioli, C., Silvestri, S., 2012. Sensor activation and radius adaptation (sara) in heterogeneous sensor networks. *ACM Trans. Sen. Netw.* 8 (3), 24:1–24:34.
- Bodor, R., Drenner, A., Schrater, P., Papanikolopoulos, N., 2007. Optimal camera placement for automated surveillance tasks. *J. Intell. Robot. Syst.* 50 (3), 257–295.
- Brown A.P., Sullivan K.J., Miller D.J., 2006. Feature-aided multiple target tracking in the image plane, vol. 6229, pp. 62290Q–62290Q–12, <http://dx.doi.org/10.1117/12.666143>.
- Bullo F., Frazzoli E., Pavone M., Savla K., Smith S.L., 2011. Dynamic vehicle routing for robotic systems. *IEEE Proc.* 99 (9), 1482–1504.
- Cannata, G., Sgorbissa, A., 2011. A minimalist algorithm for multirobot continuous coverage. *IEEE Trans. Robot.* 27 (2), 297–312.
- Cardei M., Thai M., Li Y., Wu W., 2005a. Energy-efficient target coverage in wireless sensor networks. In: INFOCOM 2005. Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1976–1984.
- Cardei M., Wu J., Lu M., Pervaiz M., 2005b. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2005 (WiMob2005), vol. 3, pp. 438–445.
- Chung W., Crespi V., Cybenko G., Jordan A., 2005. Distributed sensing and uav scheduling for surveillance and tracking of unidentifiable targets. In: Proceedings of the SPIE, vol. 5778, pp. 226–235.
- Dasgupta K., Kukreja M., Kalpakis K., 2003. Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. In: Proceedings of the Eighth IEEE International Symposium on Computers and Communication. IEEE, Kemer-Antalya, Turkey, pp. 341–348.
- Dhawan A., Vu C., Zelikovsky A., Li Y., Prasad S., 2006. Maximum lifetime of sensor networks with adjustable sensing range. In: Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006, pp. 285–289.
- Di Puglia Pugliese, L., Guerriero, F., Zorbas, D., Razafindralambo, T., 2015. Modelling the mobile target covering problem using flying drones. *Optim. Lett.*, 1–32.
- Dogan A., Zengin U., 2006. Unmanned aerial vehicle dynamic-target pursuit by using probabilistic threat exposure map. *J. Guid. Control Dyn.* 29.
- Ercan A., Yang D., Gamal A., Guibas L., 2006. Optimal placement and selection of camera network nodes for target localization. In: Gibbons P., Abdelzaher T., Aspnes J., Rao R. (Eds.), Distributed Computing in Sensor Systems. Lecture Notes in Computer Science, vol. 4026. Springer, Berlin, Heidelberg, pp. 389–404.
- González-Banos H., 2001. A randomized art-gallery algorithm for sensor placement. In: Proceedings of the Seventeenth Annual Symposium on Computational Geometry, SCG '01. ACM, New York, NY, USA, pp. 232–240.
- Gu, G., Chandler, P., Schumacher, C., Sparks, A., Pachter, M., 2006. Optimal cooperative sensing using a team of uavs. *IEEE Trans. Aerosp. Electron. Syst.* 42 (4), 1446–1458.
- Hentenryck, P., Bent, R., Upfal, E., 2010. Online stochastic optimization under time constraints. *Ann. Oper. Res.* 177 (1), 151–183.
- Hrabar S., 2008. 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pp. 807–814.
- Inaba M., Katoh N., Imai H., 1994. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In: Proceedings of the Tenth Annual Symposium on Computational Geometry, SCG '94. ACM, New York, NY, USA, pp. 332–339.
- Jin Y., Minai A., Polycarpou M., 2003. Cooperative real-time search and task allocation in uav teams. In: Proceedings of the 42nd IEEE Conference on Decision and Control, 2003, vol. 1, pp. 7–12.
- Kar K., Banerjee S., et al., 2003. Node placement for connected coverage in sensor networks. In: WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks.
- Kim J., Kim Y., 2008. Moving ground target tracking in dense obstacle areas using uavs. In: The 17th IFAC World Congress, vol. 17, pp. 8552–8557.
- Krajník T., Vonásek V., Fišer D., Faigl J., 2011. AR-Drone as a Platform for Robotic Research and Education. In: Proceedings of the Research and Education in Robotics – EUROBOT 2011: International Conference, Prague, Czech Republic, June 15–17, 2011. Springer, Berlin, Heidelberg, pp. 172–186.
- Martínez, S., Bullo, F., 2006. Optimal sensor placement and motion coordination for target tracking. *Automatica* 42 (4), 661–668.
- Müller M., Multicopter calculator, (<http://www.ecalc.ch/coptercalc.php?ecalc&lang=en>).
- Navarro-Serment L., Dolan J., Khosla P., 2004. Optimal sensor placement for cooperative distributed vision. In: 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, vol. 1, pp. 939–944.
- Oyman E., Ersoy C., 2004. Multiple sink network design problem in large scale wireless sensor networks. In: 2004 IEEE International Conference on Communications, vol. 6, pp. 3663–3667.
- Pavone, M., Savla, K., Frazzoli, E., 2009. Sharing the load. *IEEE Robot. Autom. Mag.* 16 (2), 52–61.
- Pavone, M., Arsie, A., Frazzoli, E., Bullo, F., 2011. Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Trans. Autom. Control* 56 (8), 1834–1848.
- Qi, Y., Zhao, Y., 2005. Energy-efficient trajectories of unmanned aerial vehicles flying through thermals. *J. Aerosp. Eng.* 18 (2), 84–92.
- Razafindralambo T., Mitton N., Viana A., Dias de Amorim M., Obraczka K., 2010. Adaptive deployment for pervasive data gathering in connectivity-challenged environments. In: 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 51–59.
- Ridley M., Nettleton E., Gktogan A., Brooker G., Sukkarieth S., Durrant-Whyte H., 2003. Decentralised ground target tracking with heterogeneous sensing nodes on multiple uavs. In: Zhao F., Guibas L. (Eds.), Information Processing in Sensor Networks. Lecture Notes in Computer Science, Vol. 2634. Springer, Berlin, Heidelberg, pp. 545–565.
- Schumacher C., 2005. Ground moving target engagement by cooperative uavs. In: Proceedings of American Control Conference, pp. 4502–4505.
- Simi S., Kurup R., Rao S., 2013. Distributed task allocation and coordination scheme for a multi-uav sensor network. In: 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN), pp. 1–5.
- Sinha A., Kirubakaran T., Bar-Shalom Y., 2004. Optimal cooperative placement of gmti uavs for ground target tracking. In: Proceedings of the 2004 IEEE Aerospace Conference, vol. 3, p. 1868.
- Toth, P., Vigo, D., 2001. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Wang L., Zhu H., Shen L., 2010. Cooperative ground moving target standoff tracking using uavs. In: 2010 The 2nd International Conference on Computer and

- Automation Engineering (ICCAE), vol. 2, pp. 377–382.
- Wang Q., Xu K., Takahara G., Hassanein H., 2006. Deployment for information oriented sensing coverage in wireless sensor networks. In: IEEE Global Telecommunications Conference, (GLOBECOM), pp. 1–5.
- Watanabe Y., Lesire C., Piquereau A., Fabiani P., Sanfourche M., Besnerais G.L., 2010. The onera resac unmanned autonomous helicopter: visual air-to-ground target tracking in an urban environment. In: American Helicopter Society 66th Annual Forum (AHS Forum), May.
- Xu, S., Freund, R., Sun, J., 2003. Solution methodologies for the smallest enclosing circle problem. *Comput. Optim. Appl.* 25, 283–292.
- Younis, M., Akkaya, K., 2008. Strategies and techniques for node placement in wireless sensor networks: a survey. *Ad Hoc Netw.* 6 (4), 621–655.
- Zengin, U., Dogan, A., 2011. Cooperative target pursuit by multiple uavs in an adversarial environment. *Robot. Auton. Syst.* 59 (12), 1049–1059.
- Zhou, Z., Das, S.R., Gupta, H., 2009. Variable radii connected sensor cover in sensor networks. *ACM Trans. Sen. Netw.* 5 (1), 8:1–8:36.
- Zhu, S., Wang, D., Low, C., 2013. Ground target tracking using uav with input constraints. *J. Intell. Robot. Syst.* 69 (1–4), 417–429.
- Zorbas D., Razafindralambo T., 2015. Modeling the Power Consumption of a wifibot and Studying the Role of Communication Cost in Operation Time. Technical Report. Inria Lille – Nord Europe, November.
- Zorbas D., Razafindralambo T., Di Puglia Pugliese L., Guerriero F., 2013. Energy efficient mobile target tracking using flying drones. In: The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), vol. 19, pp. 80–87.